Task armonici

- Insieme di *task armonici*: data una coppia di task qualsiasi dell'insieme, il periodo di un task è multiplo del periodo dell'altro
- Sono detti anche insiemi di task *semplicemente periodici*

- Teorema (Lehoczky et al. 1989):

  Un insieme di task armonici indipendenti, con preemption ovunque consentita e le cui deadline coincidono con il periodo, è schedulabile dall'algoritmo RM se e solo se la loro utilizzazione totale non supera 1
- Ovvero: Per insiemi di task armonici, $U_{lub}(RM)=1$

U_{lub} per task armonici

- Teorema: Per un insieme di task armonici $U_{lub}(RM)=1$.
- Dim. per contraddizione: Per un insieme di task armonici con $U \leq 1$, schedulato in modo RM, un task manca la deadline
  - Ipotizziamo che τ_i manchi la propria deadline all'istante t; t è un multiplo intero di T_i e di tutti i T_k con $k \leq i-1$
  - Tempo totale per completare i job con deadline entro t:

$$\Sigma_{k=1,i} (t/T_k)\, C_k = t\, \Sigma_{k=1,i}\, C_k/T_k = t\, U_i$$

  - ove U_i è il fattore di utilizzazione richiesto dai primi *i* task a più alta priorità e (t/T_k) è il numero di attivazioni di ciascun task τ_k entro il periodo t
  - Se il job manca la propria deadline ➔ il tempo totale richiesto era insufficiente: $t\, U_i > t \rightarrow U_i > 1 \rightarrow U > 1$ (c.v.d.)

Insiemi di task armonici - Esempio

$\tau 1$: T1 = 10 C1 = 3 U1 = 0.300

$\tau 2$: T2 = 30 C2 = 2 U2 = 0.067 U1+U2 = 0.367

$\tau 3$: T3 = 30 C3 = 5 U2 = 0.167 U1+U2+U3 = 0.534

$\tau 4$: T4 = 300 C4 = 100 U4 = 0.334 U1+U2+U3+U4 = 0.868

- L'insieme di task ($\tau 1$, $\tau 2$, $\tau 3$) é certamente schedulabile, perché U1+U2+U3 < U*(3) = 0.779
- Viceversa: U1+U2+U3+U4 > U*(4) = 0.756
- Tuttavia non é necessario operare alcuna analisi dell'intervallo critico di $\tau 4$ perché i task sono armonici

Insiemi di task armonici

- Talvolta esiste un margine di flessibilità nello specificare i periodi dei task, specie di quelli a minore priorità

- Conviene, se possibile, rendere i task armonici, per conseguire $U = 1$

- Può essere utile anche rendere armonici solo *alcuni* task, quelli per i quali esiste margine di flessibilità (per semplificare analisi e dispatching, …)

Test di schedulabilità dipendenti dai parametri dei task

- Il test basato sull'utilizzazione schedulabile è particolarmente efficiente e può essere integrato in un modulo di accettazione
- L'elaborazione è garantita se (solo sufficiente):

$$U_p \leq n(2^{1/n} - 1)$$

- Il test di Liu e Layland per RM, basato esclusivamente *sul numero* di task periodici, è solo sufficiente e risulta spesso troppo prudente
- Può essere migliorato, in alcune situazioni, considerando anche *i parametri* dei task schedulati

Test di Kuo e Mok ('91)

- Un insieme S di N task periodici con $D_i=T_i$ è schedulabile da RM se può essere *partizionato* in N_h sottoinsiemi disgiunti S_i tali che all'interno di ogni partizione i task risultano *semplicemente periodici*, ed è schedulabile da RM l'insieme di task S' in cui ad ogni partizione corrisponde un task di periodo minimo tra i task della partizione e fattore di utilizzazione pari alla somma delle utilizzazioni dei task della partizione.
- Uh?

- Ovvero: $U_{lub}(RM) = N_h(2^{1/N_h} - 1)$

Test di Kuo e Mok - esempio

- Set iniziale: $\tau_1=(10,5)$, $\tau_2=(25,5)$, $\tau_3=(50,5)$
- $U1=0.5$, $U2=0.2$, $U3=0.1$, $\Sigma Uj=0.8 > U_{LL}(3)=0.78$

- Partizione: $(\tau_1),(\tau_2, \tau_3)$
- Set trasformato:

  $\tau'_1=(T=10, U=0.5)$, $\tau'_2=(T=25, U=0.3)$

- Per il set trasformato vale $N_h=2$: $\Sigma Uj=0.8 < U_{LL}(N_h=2)=0.828$
- → il set <u>iniziale</u> è garantito dal bound KM se schedulato in modo RM
- In generale è sufficiente verificare che $U \leq N_h(2^{1/Nh} - 1)$

Applicazione del test di Kuo e Mok

- In generale, per applicare il bound di Kuo-Mok *non è necessario costruire alcun set equivalente*!
- E' sufficiente calcolare l'utilizzazione richiesta e confrontare tale valore con il bound modificato tenendo conto del teorema di Kuo e Mok
- L'utilizzazione richiesta non cambia applicando il bound di Liu e Layland o quello di Kuo e Mok (né altri bound)!
- In altre parole: occorre sempre considerare i task con priorità rate-monotonic decrescente applicando il bound di Kuo e Mok *al sottoinsieme di task in esame*: <u>lo scheduling resta rate-monotonic</u>

Test di Kuo e Mok – variazione esempio

- Set iniziale: τ_1=(10,6), τ_2=(25,5), τ_3=(50,5)
- U1=0.6, U2=0.2, U3=0.1, ΣUj=0.9 > $U_{LL}(3)$=0.78

- Partizione: $P_a=\{(\tau_1,\tau_3),(\tau_2)\}$ oppure $P_b=\{(\tau_1),(\tau_2,\tau_3)\}$
  - set trasformato P_a: τ'_1=(T=10, U=0.7), τ'_2=(T=25, U=0.2)
  - set trasformato P_b: τ'_1=(T=10, U=0.6), τ'_2=(T=25, U=0.3)

- Per il set trasformato, N_h=2: ΣUj=0.9 > $U_{LL}(2)$=0.828

  → il set <u>iniziale</u> non è garantito dal bound KM
- Quali task sono individualmente garantiti?
- Ricordarsi: *lo scheduling resta RM*!

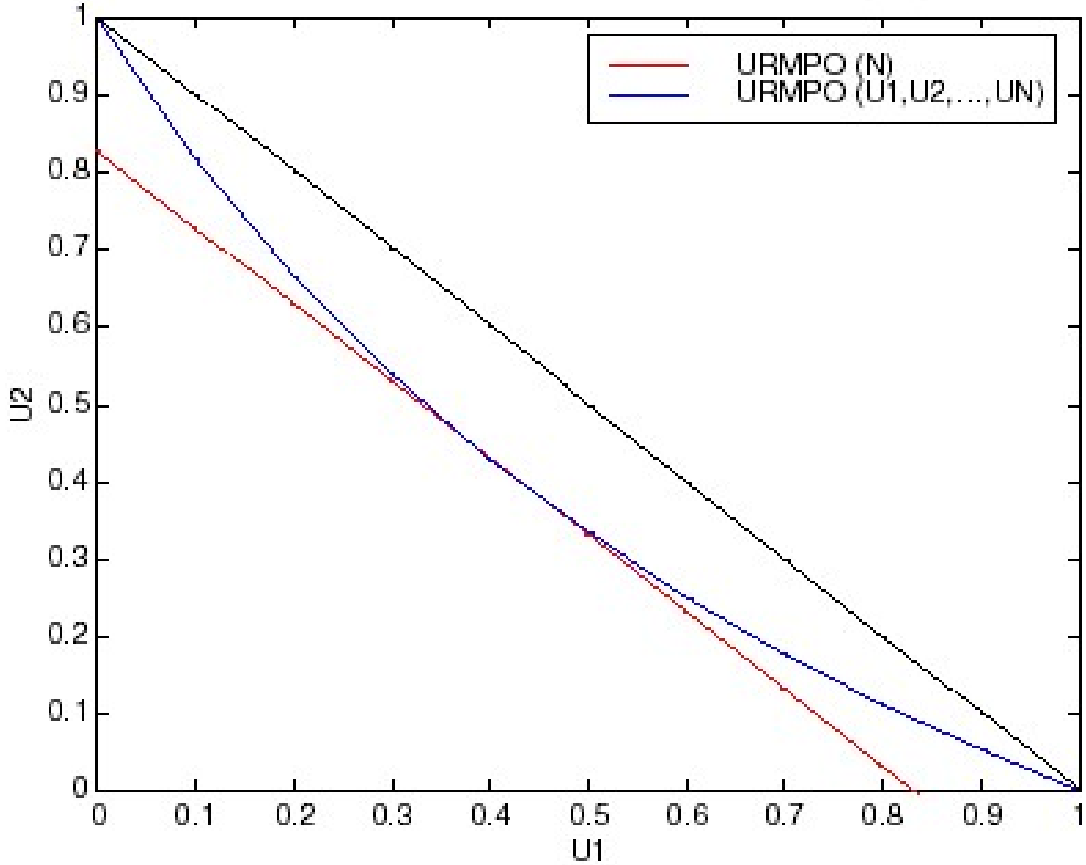
Il bound iperbolico

- Hyperbolic Bound (HB) è un bound meno restrittivo sul fattore di utilizzazione disponibile, applicabile per l'algoritmo RM
- Come il bound di Kuo e Mok, richiede di esplicitare *i parametri temporali* dei task periodici
- Meno conservativo del bound di Liu e Layland (che si basa *solo sul numero di task*)
- Un insieme di N task eseguito con priorità RM è garantito se vale:

$$\prod_{(j=1,N)} (1 + U_j) \leq 2$$

Il bound iperbolico

- Per 2 task:



Bound iperbolico - Esempio

- Set iniziale: $\tau_1=(10,5)$, $\tau_2=(25,5)$, $\tau_3=(50,5)$
- U1=0.5, U2=0.2, U3=0.1, ΣUj=0.8 > $U_{LL}(3)=0.78$

- Vale: $\Pi_{(j=1,3)} (1 + U_j) = 1.98 \leq 2$

- Poiché il bound iperbolico è soddisfatto, l'insieme di task è garantito (dal bound iperbolico) se schedulato in modo RM

- Esercizio: $\tau_1=(10,5)$, $\tau_2=(25,5)$, $\tau_3=(55,6)$

Algoritmo Deadline Monotonic

- Algoritmo DM (Leung e Whitehead, 1982):
  ordina le priorità in funzione delle *deadline relative* D_i

- *Ottimalità di DM*: Se un insieme di task è schedulabile da un algoritmo a priorità fissa, esso può essere schedulato anche dall'algoritmo DM

- DM opera in *ipotesi più generali* di RM: con *deadline relative arbitrarie* DM è ottimo, mentre RM non lo è
  - ovvero: RM resta ottimo solo se le deadline relative sono proporzionali ai periodi

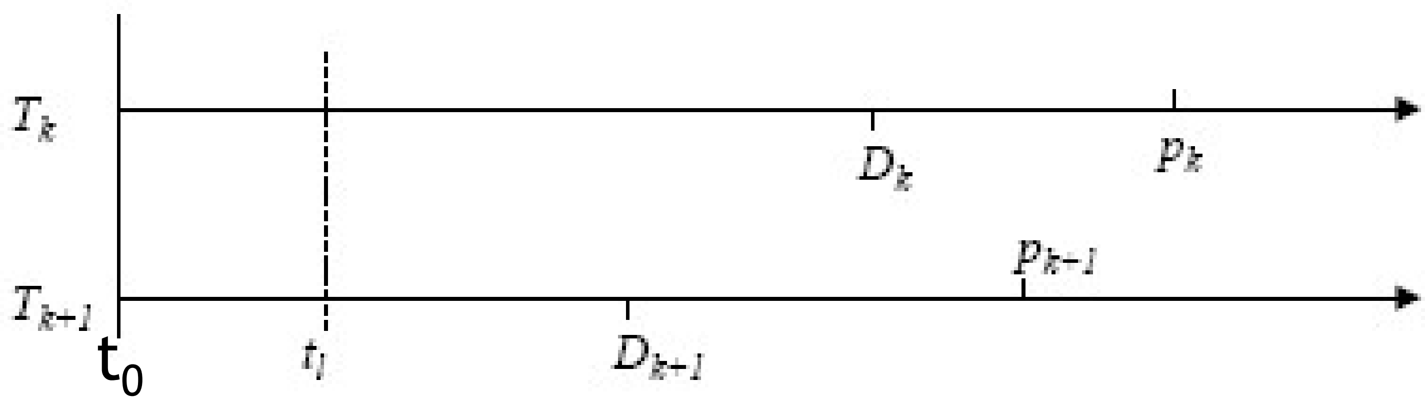
Algoritmo Deadline Monotonic

- Dimostrazione dell'ottimalità di DM:

- Insieme di task $\tau_1, \tau_2, \ldots, \tau_n$ a priorità decrescente, schedulabile in modo non DM. Esiste quindi una coppia di task consecutivi τ_k e τ_{k+1} per i quali $D_k > D_{k+1}$
- Consideriamo l'intervallo critico a partire da t_0 di τ_{k+1}: è la situazione peggiore per tutti gli algoritmi a priorità statica. Valutiamo l'orizzonte temporale fino a D_{k+1} ($<D_k$)
- Poiché l'insieme è schedulabile in modo non DM, τ_{k+1} completa entro D_{k+1} ed è preceduto dai job a più alta priorità ed anche dal job (unico nell'intervallo critico) di τ_k

Algoritmo Deadline Monotonic

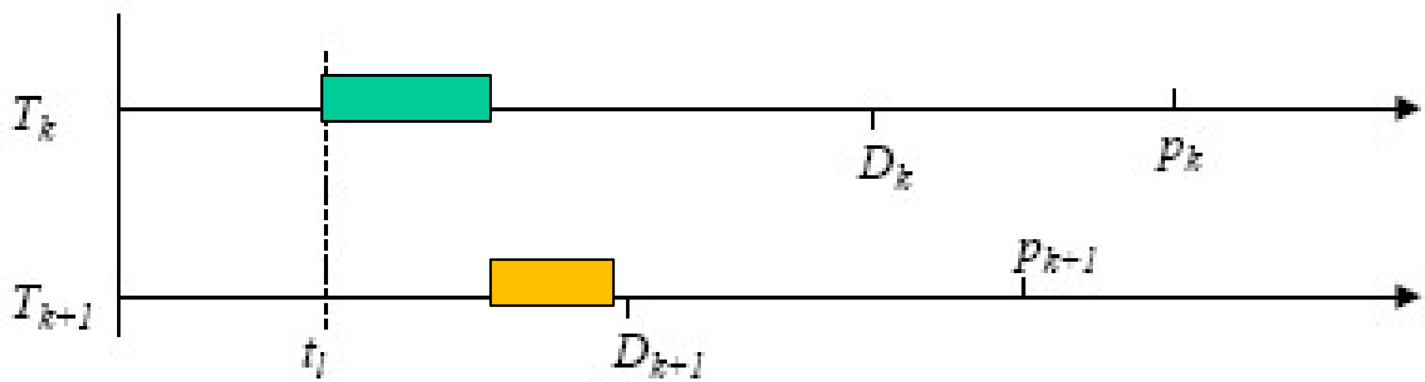
- L'esecuzione di τ_k e τ_{k+1} può essere divisa in più intervalli per le preemption da parte di job a più alta priorità, ma in ogni caso scambiando la priorità di τ_k e τ_{k+1}, e quindi l'ordine di esecuzione negli intervalli che precedono D_{k+1}, l'insieme resta schedulabile



Algoritmo Deadline Monotonic

- Iterando tra tutte le coppie di task non ordinate con priorità DM si completa la dimostrazione

- Nella situazione in figura, scambiando in t_l la priorità tra τ_k e τ_{k+1} l'insieme di task resta schedulabile



Utilizzazione schedulabile per DM

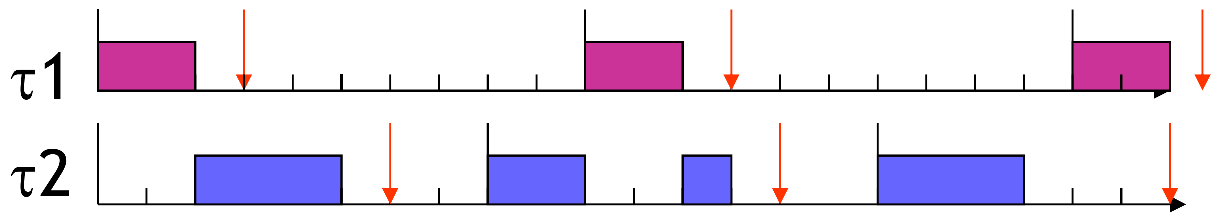
- Per garantire un insieme di *n* task periodici *con deadline relativa minore o uguale del periodo* $D_i \leq T_i \; \forall i$, si può utilizzare una variante del test RM:

$$\Sigma_{i=1,n} C_i/D_i \leq n(2^{1/n} - 1) \qquad (*)$$

- Questo condizione è solo sufficiente e in generale è *estremamente peggiorativa*
- Se $D_i < T_i$ per qualche i, un insieme di task potrebbe essere schedulabile anche con $\Sigma_{i=1,n} C_i/D_i > 1$
- Se il test (*) non è soddisfatto, occorre effettuare un'analisi di schedulabilità a partire dall'istante critico nell'asse del tempo

Schedulazione DM

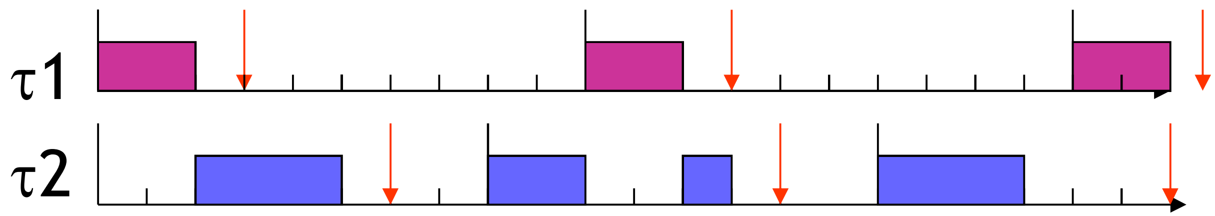
- Esempio: $\tau 1=(10, 2, 3)$ $\tau 2=(8, 3, 6)$ $[\tau_i=(T_i, C_i, D_i)]$



- $\Sigma_{i=1,n} C_i/D_i=2/3+3/6=1.16 > 1$ -- cosa rappresenta?

Non ottimalità di RM nel caso $D_i \neq T_i$

- Dimostrazione:
- Esempio: $\tau 1=(10, 2, 3)$ $\tau 2=(8, 3, 6)$ $[\tau_i=(T_i, C_i, D_i)]$



- RM non produce una schedule fattibile ($\tau 1$ manca la deadline) mentre DM sì, pertanto RM non è ottimo in queste ipotesi più generali
- $\sum_{i=1,n} C_i/T_i=2/10+3/8=0,575$ ← utilizzazione